Information Extraction

Philipp Koehn

28 October 2019

Text → Knowledge

Knowledge

From Wikipedia, the free encyclopedia

"Know" redirects here. For the Jason Mraz album, see Know (album). For other uses, see Knowledge (disambiguation).

Knowledge is a familiarity, awareness, or understanding of someone or something, such as facts, information, descriptions, or skills, which is acquired through experience or education by perceiving, discovering, or learning.

Knowledge can refer to a theoretical or practical understanding of a subject. It can be implicit (as with practical skill or expertise) or explicit (as with the theoretical understanding of a subject); it can be more or less formal or systematic.[1] In philosophy, the study of knowledge is called epistemology; the philosopher Plato famously defined knowledge as "justified true belief", though this definition is now thought by some analytic philosophers[citation needed] to be problematic because of the Gettier problems, while others defend the platonic definition.[2] However, several definitions of knowledge and theories to explain it exist.

Knowledge acquisition involves complex cognitive processes: perception, communication, and reasoning;[3] while knowledge is also said to be related to the capacity of *acknowledgement* in human beings.[4]

Contents [hide]

Theories of knowledge

Main article: Epistemology

The eventual demarcation of philosophy from science was made possible by the notion that philosophy's core was "theory of knowledge," a theory distinct from the sciences because it was their *foundation*... Without this idea of a "theory of knowledge," it is hard to imagine what "philosophy" could have been in the age of modern science.

— Richard Rorty, *Philosophy and the Mirror of Nature*

Can You Really Be Addicted to Video Games?

The latest research suggests it's not far-fetched at all — especially when you consider all the societal and cultural factors that make today's games so attractive.

By Ferris Jabr

Published Oct. 22, 2019
Updated Oct. 23, 2019, 3:18 p.m. ET

83

Charlie Bracke can't remember a time when he wasn't into video games. When he was 5, he loved playing Wolfenstein 3D, a crude, cartoonish computer game in which a player tries to escape a Nazi prison by navigating virtual labyrinths while mowing down enemies. In his teenage years, he became obsessed with more sophisticated shooters and a new generation of online games that allowed thousands of players to inhabit sprawling fantasy worlds. Ultima Online, World of Warcraft, The Elder Scrolls — he would spend as much as 12 hours a day in these imaginary realms, building cities and fortifications, fighting in epic battles and hunting for treasure.

During his childhood, Bracke's passion for video games, like that of most young Americans, didn't cause him any serious problems. At school, he got along with just about everyone and maintained straight A's. His homework was easy enough that he could complete it on the bus or in class, which allowed him to maximize

Open Information Extraction from the Web

Michele Banko, Michael J Cafarella, Stephen Soderland, Matt Broadhead and Oren Etzioni
Turing Center
Department of Computer Science and Engineering
University of Washington
Box 352350
Seattle, WA 98195, USA
{banko,mjc,soderlan,hastur,etzioni}@cs.washington.edu

Abstract

Traditionally, Information Extraction (IE) has focused on satisfying precise, narrow, pre-specified requests from small homogeneous corpora (e.g., extract the location and time of seminars from a set of announcements). Shifting to a new domain requires the user to name the target relations and to manually create new extraction rules or hand-tag new training examples. This manual labor scales linearly with the number of target relations.

This paper introduces *Open IE* (OIE), a new extraction paradigm where the system makes a single data-driven pass over its corpus and extracts a large set of relational tuples without requiring *any* human input. The paper also introduces TEXTRUNNER, a fully implemented, highly scalable OIE system where the tuples are assigned a probability and indexed to support efficient extraction and exploration via user queries.

We report on experiments over a 9,000,000 Web page corpus that compare TEXTRUNNER with KNOWITALL, a state-of-the-art Web IE system. TEXTRUNNER achieves an error reduction of 33% on a comparable set of extractions. Furthermore, in the amount of time it takes KNOWITALL to perform extraction for a handful of pre-specified relations, TEXTRUNNER extracts a far broader set of facts reflecting orders of magnitude more relations, discovered on the fly. We report statistics on TEXTRUNNER's 11,000,000 highest probability tuples, and show that they contain over 1,000,000 concrete facts and over 6,500,000 more abstract assertions.

1 Introduction and Motivation

This paper introduces *Open Information Extraction (OIE)*—a novel extraction paradigm that facilitates domain-independent discovery of relations extracted from text and readily scales to the diversity and size of the Web corpus. The sole input to an OIE system is a corpus, and its output is a set of extracted relations. An OIE system makes a single pass over its corpus guaranteeing scalability with the size of the corpus.

Information Extraction (IE) has traditionally relied on extensive human involvement in the form of hand-crafted extraction rules or hand-tagged training examples. Moreover, the user is required to explicitly pre-specify each relation of interest. While IE has become increasingly automated over time, enumerating all potential relations of interest for extraction by an IE system is highly problematic for corpora as large and varied as the Web. To make it possible for users to issue diverse queries over heterogeneous corpora, IE systems must move away from architectures that require relations to be specified prior to query time in favor of those that aim to discover all possible relations in the text.

In the past, IE has been used on small, homogeneous corpora such as newswire stories or seminar announcements. As a result, traditional IE systems are able to rely on "heavy" linguistic technologies tuned to the domain of interest, such as dependency parsers and Named-Entity Recognizers (NERs). These systems were not designed to scale relative to the size of the corpus or the number of relations extracted, as both parameters were fixed and small.

The problem of extracting information from the Web violates all of these assumptions. Corpora are massive and heterogeneous, the relations of interest are unanticipated, and their number can be large. Below, we consider these challenges in more detail.

Automation The first step in automating IE was moving from knowledge-based IE systems to trainable systems that took as input hand-tagged instances [Riloff, 1996] or document segments [Craven *et al.*, 1999] and automatically learned domain-specific extraction patterns. DIPRE [Brin, 1998], SNOWBALL [Agichtein and Gravano, 2000], and Web-based question answering systems [Ravichandran and Hovy, 2002] further reduced manual labor needed for relation-specific text extraction by requiring only a small set of tagged seed instances or a few hand-crafted extraction patterns, per relation, to launch the training process. Still, the creation of suitable training data required substantial expertise as well as non-trivial manual effort *for every relation extracted*, and the relations have to be specified in advance.

Corpus Heterogeneity Previous approaches to relation extraction have employed kernel-based methods [Bunescu

- Human knowledge is stored in text
- How can we extract this to make it available for processing by machines?

examples

Goal: Build Database of World Leaders

Country	Position	Person
United States	president	George Walker Bush
United States	president	Barack Hussein Obama
United States	president	Donald Trump
Germany	chancellor	Gerhard Schröder
Germany	chancellor	Angela Merkel
United Kingdom	prime minister	Theresa May
United Kingdom	prime minister	Alexander Boris de Pfeffel Johnson
China	president	Hu Jintao
China	president	Xi Jinping
India	prime minister	Manmohan Singh
India	prime minister	Narendra Modi

Extracting Relations

Barack Hussein Obama was elected the 44th president of the United States on Tuesday, sweeping away the last racial barrier in American politics with ease as the country chose him as its first black chief executive.

- From this snippet, we can extract:

```
(United States, president, Barack Hussein Obama)
```

- Why is this a hard problem?

Extracting Events

Serge Gnabry scores four in brutal Bayern's 7-2 humiliation of Tottenham

There were 55 minutes on the stadium clock and a look of pure bewilderment on the faces of Tottenham's defenders. They had just been shredded for the second time in three minutes by Serge Gnabry, Bayern Munich's former Arsenal winger and, despite having carried the fight to the Bundesliga champions, they were staring at an irretrievable 4-1 deficit.

Irresistible Gnabry shreds Spurs after Lewandowski's decisive incision

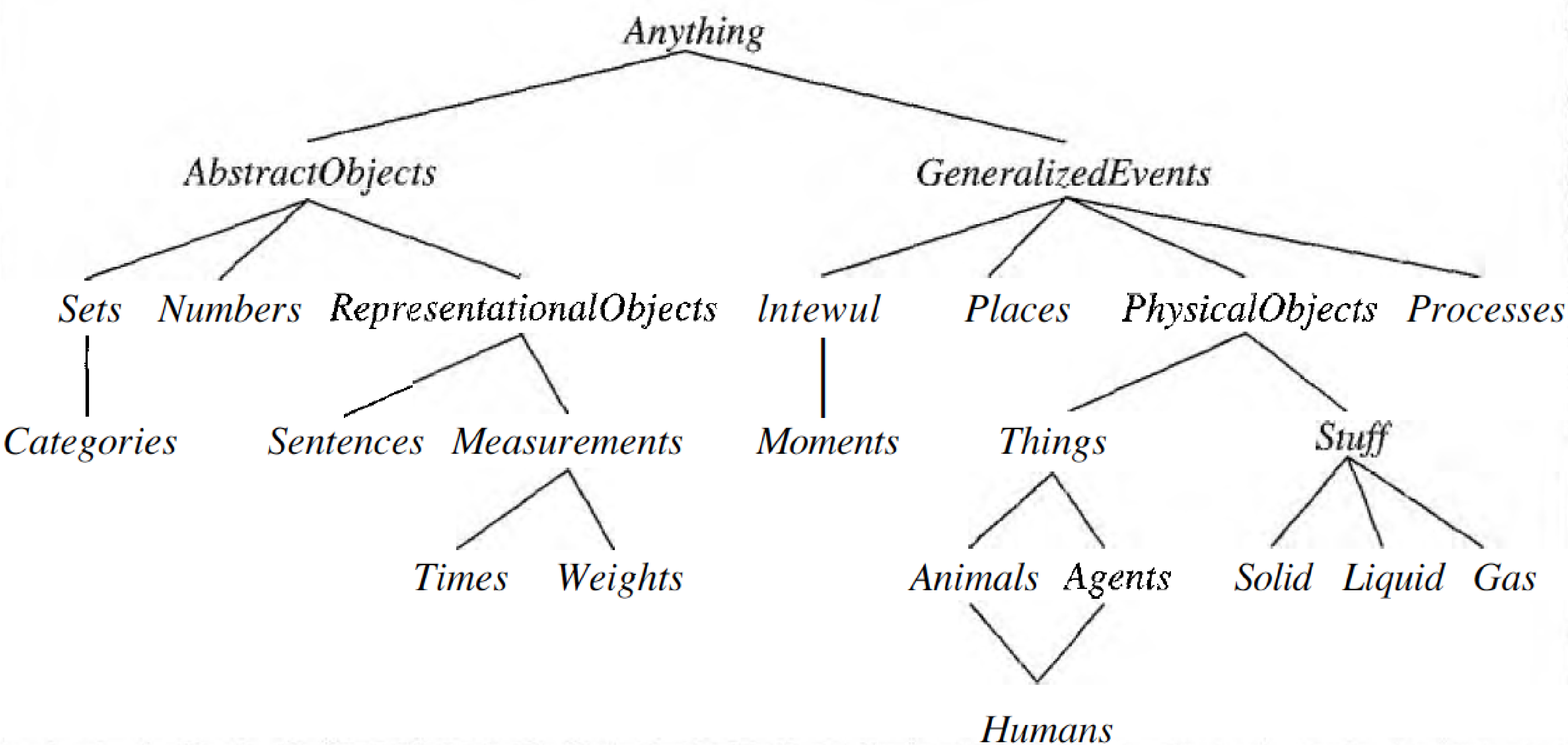
Read more

Remarkably, it was only the beginning of the pain for the manager, Mauricio Pochettino, and his team on a night when serious questions could be asked of their character. It has been a difficult season for them so far, with off-field issues from the summer hanging over into the opening weeks. Pochettino has repeatedly given the impression that he has been fighting with one hand tied behind his back. Here, he could do nothing to stop the onslaught as Gnabry and Bayern twisted the knife.

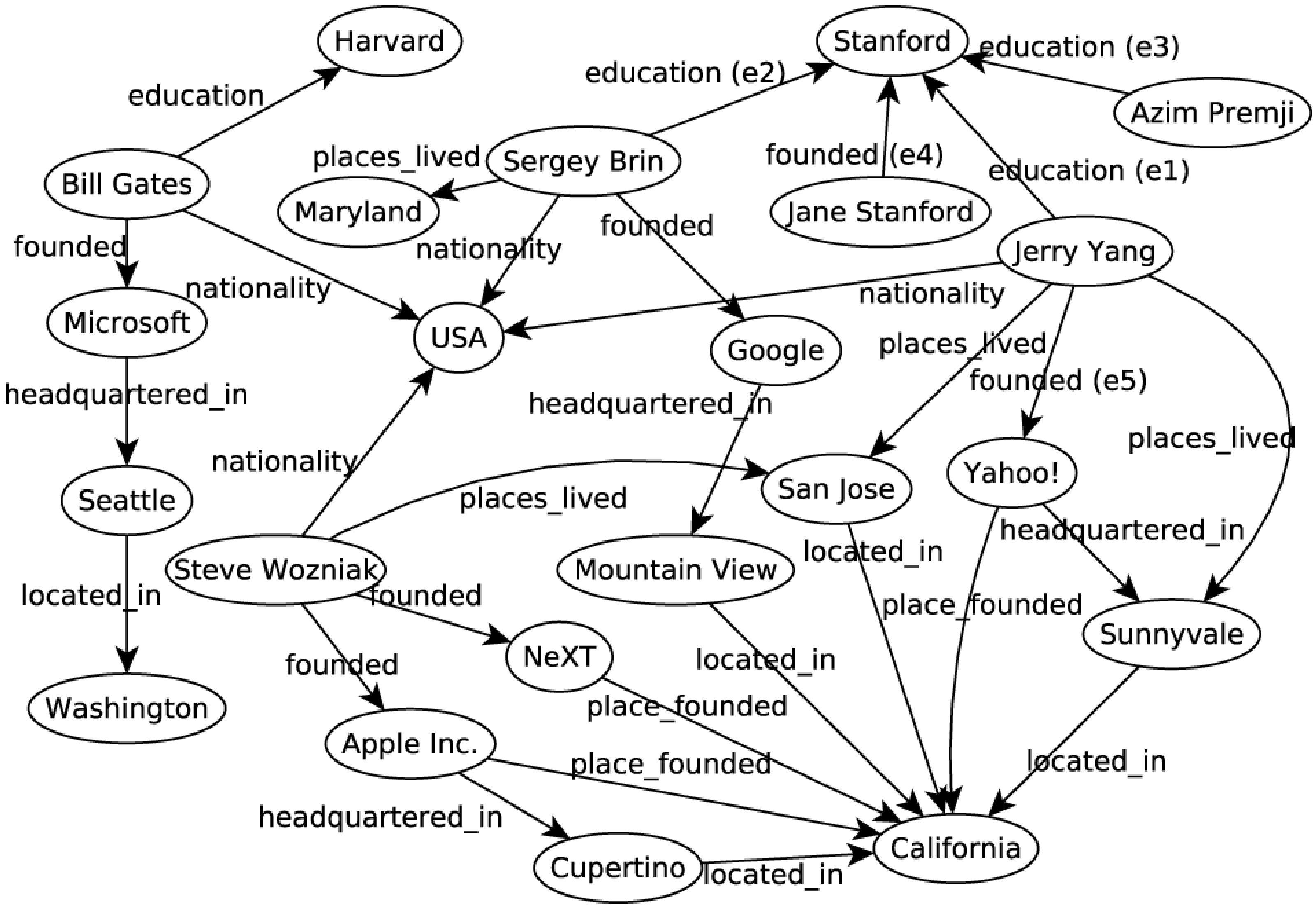
- Report of soccer game
  - when? where? who? what? why?
  - players involved, information about each player, each goal, audience size, ...?
- Multiple data base tables, connection between entities

structural knowledge

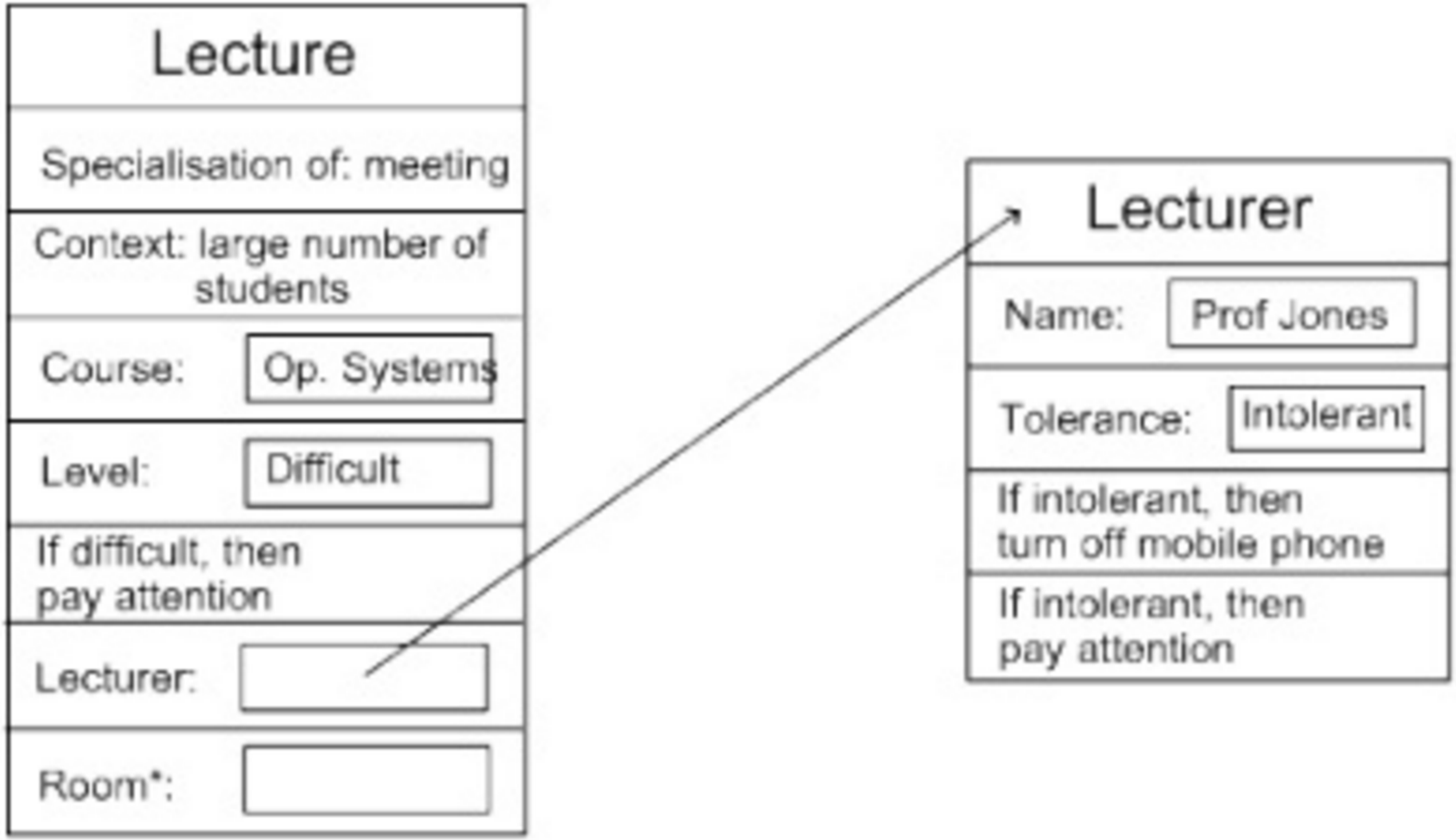
Ontologies


Anything
AbstractObjects
GeneralizedEvents
Sets
Numbers
RepresentationalObjects
lntewul
Places
PhysicalObjects
Processes
Categories
Sentences
Measurements
Moments
Things
Stuff
Times
Weights
Animals
Agents
Solid
Liquid
Gas
Humans

Knowledge Graphs



Frames


Lecture
Specialisation of: meeting
Context: large number of students
Course:
Op. Systems
Level:
Difficult
If difficult, then pay attention
Lecturer:
Room*:
Lecturer
Name:
Prof Jones
Tolerance:
Intolerant
If intolerant, then turn off mobile phone
If intolerant, then pay attention

Scripts

Script Restaurant

Props
- •Tables
- •Menu
- •F = Food
- •Check
- •Money

Roles
- •P = Customer
- •O = Waiter
- •V = Cook
- •K = Cashier
- •S = Owner

Entry conditions
- •P is hungry
- •P has money

Results
- •P has less money
- •P is not hungry
- •P is pleased (optional)
- •S has more money

Scene 1: Entering
P PTRANS P into restaurant
P ATTEND eyes to tables
P MBUILD where to sit
P PTRANS P to table
P MOVE P to sitting position

Scene 2: Ordering
(Menu on table)
O brings menu)
P PTRANS menu to P

(S asks for menu)
S MTRANS signal to O
O PTRANS O to table
P MTRANS "need menu" to O
O PTRANS O to menu

O PTRANS O to table
O ATRANS menu to P

P MTRANS food list to P
* P MBUILD choice of F
P MTRANS signal to O
O PTRANS O to table
P MTRANS 'I want F' to O

O PTRANS O to V
O MTRANS (ATRANS F) to V

V MTRANS 'no F' to O
O PTRANS O to P
O MTRANS 'no F' to P
(go back to *) or
(go to Scene 4 at no pay path)

V DO (prepare F script)
to Scene 3

Scene 3: Eating
V ATRANS F to O
O ATRANS F to P
P INGEST F

Option: Return to Scene 2 to order more; otherwise, go to Scene 4

Scene 4: Exiting
P MTRANS to O
(O ATRANS check to P)

O MOVE write check
O PTRANS O to P
O ATRANS check to P
P ATRANS tip to O
P PTRANS P to K
P ATRANS money to K
P PTRANS P to out of restaurant

No pay path

Schank un Abelson, 1977

named entities

Named Entities

- Essential processing step: identifying named entities
- Types
  - persons
  - geo-political entities (GPE)
  - events
  - dates
  - numbers

Example

[PERSON Boris Johnson]'s [GPE cabinet] is divided over how to proceed with [EVENT Brexit], as the [PERSON prime minister] faces the stark choice of pressing ahead with his deal or gambling his premiership on a [DATE pre-Christmas] general election. The [PERSON prime minister] told [PERSON MPs] at [DATE Wednesday]'s [EVENT PMQs] that he was awaiting the decision of the [GPE EU27] over whether to grant an extension before settling his next move. Some [PERSON cabinet ministers], including the [PERSON [GPE Northern Ireland] secretary, Julian Smith], believe the majority of [NUMBER 30] achieved by the [GPE government] on the second reading of the [EVENT Brexit] bill on [DATE Tuesday] suggests [PERSON Johnson]'s deal has enough support to carry it through all its stages in [GPE parliament].

Named Entity Tagging

- Problem broken up into two parts

- Tagging where named entities start and end

  `[NE Boris Johnson]'s [NE cabinet] is divided over how to proceed with [NE Brexit], as the [NE prime minister] faces the stark`

- Classification of types

  `[PERSON Boris Johnson]'s [GPE cabinet] is divided over how to proceed with [EVENT Brexit], as the [PERSON prime minister] faces the stark`

Tagging

- Convert into BIO sequence (begin / intermediate / other)

```
Boris     B
Johnson   I
's        O
cabinet   B
is        O
divided   O
over      O
how       O
to        O
proceed   O
with      O
Brexit    B
,         O
```

Bayes Rule

- We want to find the best part-of-speech tag sequence T for a sentence S

$$\text{argmax}_T \; p(T|S)$$

Bayes Rule

- We want to find the best part-of-speech tag sequence T for a sentence S

$$\text{argmax}_T \; p(T|S)$$

- Bayes rule gives us

$$p(T|S) = \frac{p(S|T)\; p(T)}{p(S)}$$

Bayes Rule

- We want to find the best part-of-speech tag sequence T for a sentence S

$$\text{argmax}_T \; p(T|S)$$

- Bayes rule gives us

$$p(T|S) = \frac{p(S|T)\; p(T)}{p(S)}$$

- We can drop $p(S)$ if we are only interested in argmax_T

$$\text{argmax}_T \; p(T|S) = \text{argmax}_T \; p(S|T)\; p(T)$$

Decoming the Model

- The mapping $p(S|T)$ can be decomposed into

$$p(S|T) = \prod_i p(w_i|t_i)$$

Decomposing the Model

- The mapping $p(S|T)$ can be decomposed into

$$p(S|T) = \prod_i p(w_i|t_i)$$

- $p(T)$ could be called a *part-of-speech language model*, for which we can use an n-gram model:

$$p(T) = p(t_1)\ p(t_2|t_1)\ p(t_3|t_1, t_2)...p(t_n|t_{n-2}, t_{n-1})$$

Decomposing the Model

- The mapping $p(S|T)$ can be decomposed into

$$p(S|T) = \prod_i p(w_i|t_i)$$

- $p(T)$ could be called a *part-of-speech language model*, for which we can use an n-gram model:

$$p(T) = p(t_1)\, p(t_2|t_1)\, p(t_3|t_1, t_2) ... p(t_n|t_{n-2}, t_{n-1})$$

- We can estimate $p(S|T)$ and $p(T)$ with maximum likelihood estimation (and maybe some smoothing)

Hidden Markov Model (HMM)

- The model we just developed is a **Hidden Markov Model**
- Elements of an HMM model:
  - a set of states (here: the tags)
  - an output alphabet (here: words)
  - initial state (here: beginning of sentence)
  - state transition probabilities (here: $p(t_n|t_{n-2}, t_{n-1})$)
  - symbol emission probabilities (here: $p(w_i|t_i)$)

Search for the Best Tag Sequence

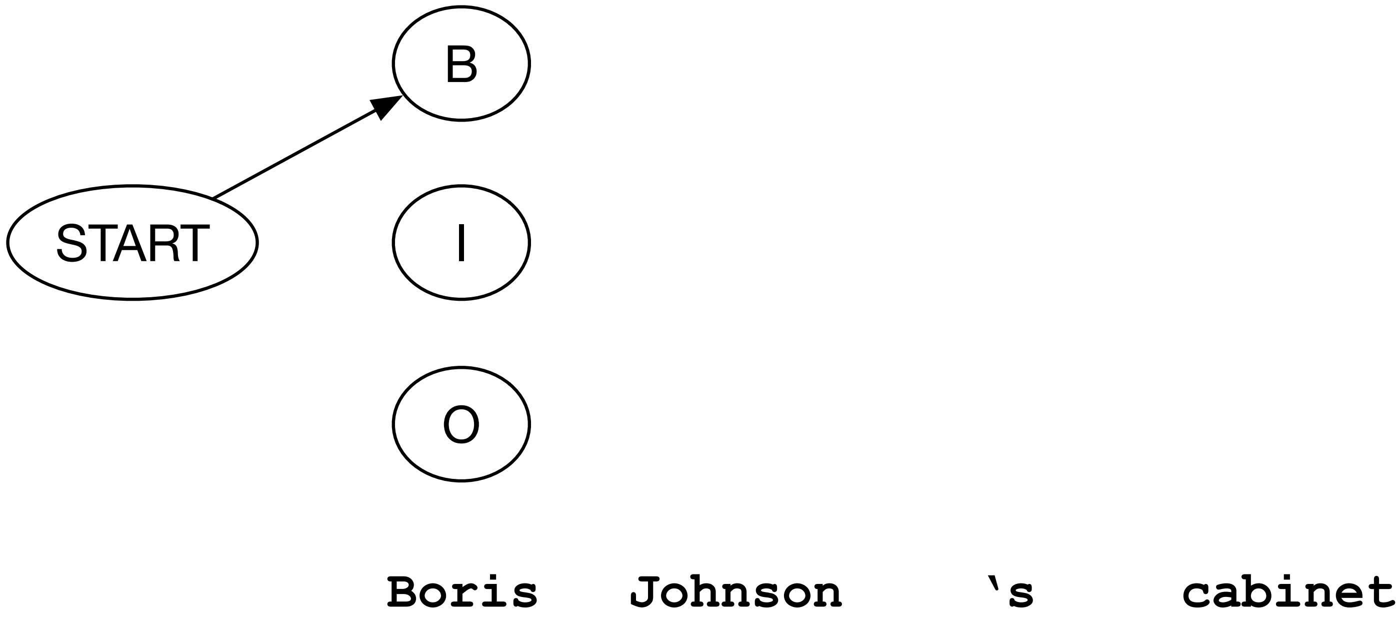
- We have defined a model, but how do we use it?
  - given: word sequence
  - wanted: tag sequence
- If we consider a specific tag sequence, it is straight-forward to compute its probability

$$p(S|T)\,p(T) = \prod_i p(w_i|t_i)\,p(t_i|t_{i-2},t_{i-1})$$

- Problem: if we have on average c choices for each of the n words, there are c^n possible tag sequences, maybe too many to efficiently evaluate

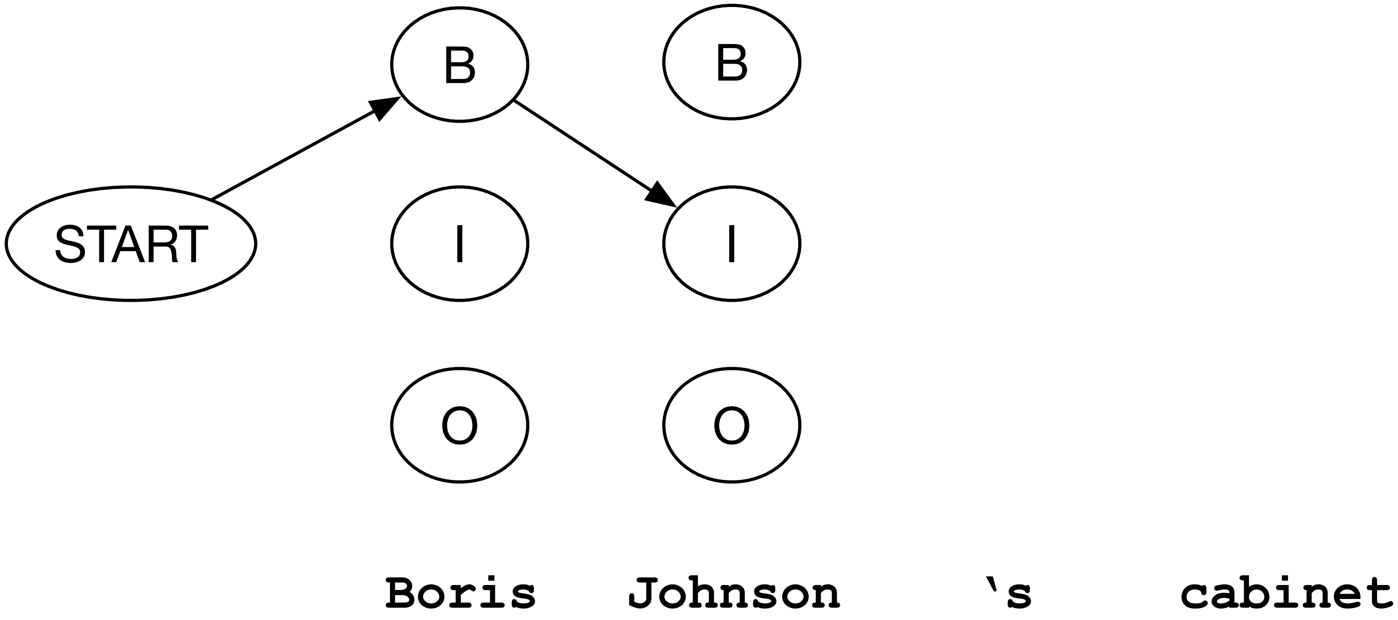
Walking through the States

- First, we go to state B to emit `Boris`:



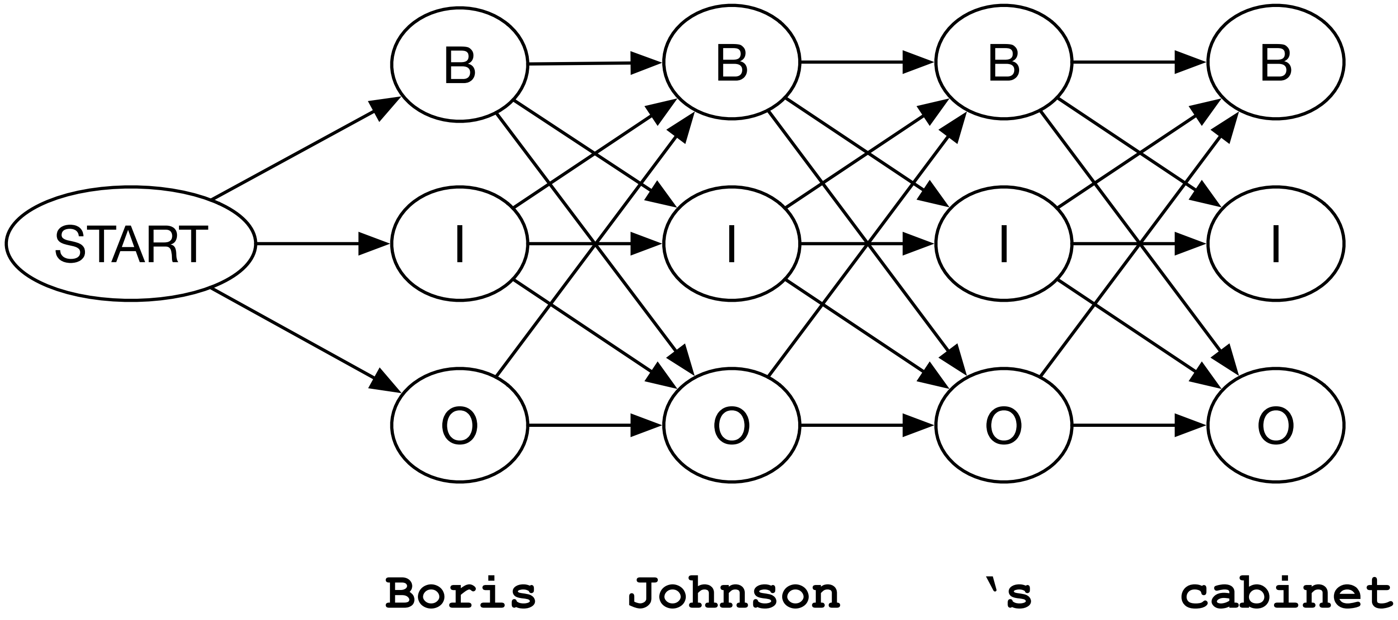
Walking through the States

- Then, we go to state *I* to emit `Johnson`:



Walking through the States

- Of course, there are many possible paths:



Viterbi Algorithm

- Intuition: Since state transition out of a state only depend on the current state (and not previous states), we can record for each state the optimal path

Viterbi Algorithm

- Intuition: Since state transition out of a state only depend on the current state (and not previous states), we can record for each state the optimal path

- We record:
  - cheapest cost to state j at step s in $\delta_j(s)$
  - backtrace from that state to best predecessor $\psi_j(s)$

Viterbi Algorithm

- Intuition: Since state transition out of a state only depend on the current state (and not previous states), we can record for each state the optimal path

- We record:
  - cheapest cost to state j at step s in $\delta_j(s)$
  - backtrace from that state to best predecessor $\psi_j(s)$

- Stepping through all states at each time steps allows us to compute
  - $\delta_j(s+1) \quad = \quad \max_{1 \leq i \leq N} \delta_i(s)\; p(t_i|t_j)\; p(w_s|t_j)$
  - $\psi_j(s+1) = \text{argmax}_{1 \leq i \leq N} \delta_i(s)\; p(t_i|t_j)\; p(w_s|t_j)$

Viterbi Algorithm

- Intuition: Since state transition out of a state only depend on the current state (and not previous states), we can record for each state the optimal path
- We record:
  - cheapest cost to state j at step s in $\delta_j(s)$
  - backtrace from that state to best predecessor $\psi_j(s)$
- Stepping through all states at each time steps allows us to compute
  - $\delta_j(s+1) \quad = \quad \max_{1 \leq i \leq N} \delta_i(s)\ p(t_i|t_j)\ p(w_s|t_j)$
  - $\psi_j(s+1) = \text{argmax}_{1 \leq i \leq N}\ \delta_i(s)\ p(t_i|t_j)\ p(w_s|t_j)$
- Best final state is $\text{argmax}_{1 \leq i \leq N}\ \delta_i(S+1)$, we can backtrack from there

entity linking

Same Person

[PERSON **Boris Johnson**]'s cabinet is divided over how to proceed with Brexit, as the [PERSON **prime minister**] faces the stark choice of pressing ahead with his deal or gambling his premiership on a pre-Christmas general election. The [PERSON **prime minister**] told MPs at Wednesday's PMQs that he was awaiting the decision of the EU27 over whether to grant an extension before settling his next move. Some cabinet ministers, including the secretary, Julian Smith, believe the majority of 30 achieved by the government on the second reading of the Brexit bill on Tuesday suggests [PERSON **Johnson**]'s deal has enough support to carry it through all its stages in parliament.

- Same person referred to 4 times in 3 different ways

Different Person, Same Name

- **Explorers and Academics**
  - John Smith (explorer) (1580–1631), helped found the Virginia Colony and became Colonial Governor of Virginia
  - John Smith (anatomist and chemist) (1721–1797), professor of anatomy and chemistry at the University of Oxford, 1766–97
  - John Smith (Cambridge, 1766), vice chancellor of the University of Cambridge, 1766 until 1767
  - John Smith (astronomer) (1711–1795), Lowndean Professor of Astronomy and Master of Caius
  - John Smith (lexicographer) (died 1809), professor of languages at Dartmouth College
  - John Smith (botanist) (1798–1888), curator of Kew Gardens
  - John Smith (physician) (c.1800–1879), Scottish physician specialising in treating the insane
  - John Smith (dentist) (1825–1910), founder of Edinburgh's School of Dentistry
  - John Smith (sociologist) (1927–2002), English sociologist
- **Arts**
  - John Smith (engraver) (1652–1742), English mezzotint engraver
  - John Smith (English poet) (1662–1717), English poet and playwright
  - John Smith (clockmaker) (1770–1816), Scottish clockmaker
  - John Smith (architect) (1781–1852), Scottish architect
  - John Smith (art historian) (1781–1855), British art dealer
  - John Smith (Canadian poet) (born 1927), Canadian poet
  - John Smith (actor) (1931–1995), American actor
  - John Smith (English filmmaker) (born 1952), avant-garde filmmaker
  - John Smith (comics writer) (born 1967), British comics writer
  - John Smith (musician), English contemporary folk musician and recording artist
- **Politicians**
  - John Smith (Victoria politician) (John Thomas Smith, 1816–1879), Australian politician
  - John Smith (New South Wales politician, born 1811) (1811–1895), Australian politician
  - John Smith (New South Wales politician, born 1821) (1821–1885), Scottish/Australian professor and politician
  - John Smith (Kent MPP), member of the 1st Ontario Legislative Assembly, 1867–1871
  - John Smith (Manitoba politician) (1817–1889), English-born farmer and politician in Manitoba
  - John Smith (Peel MPP) (1831–1909), Scottish-born Ontario businessman and political figure
- ... many many more ...

Entity Linking

- Task: map a mention to an entity

- Entity linking is often formulated as a ranking problem

$$y^* = \text{argmax}_y \; \Psi(y, x, c) \qquad\qquad y \in Y(x)$$

  where

  - y is a target entity,
  - x is a description of the mention
  - $Y(x)$ is a set of candidate entities
  - c is a description of the context
  - Ψ is a scoring function

- Predefined name dictionary to restrict set of candidates $Y(x)$

Features

- Similarity of mention string to canonical entity name

  $\Psi(\texttt{ATLANTA}, \texttt{Atlanta}, c) > \Psi(\texttt{ATLANTA} - \texttt{HAWKS}, \texttt{Atlanta}, c)$

- Popularity of the entity (e.g., measured by Wikipedia page views)

  $\Psi(\texttt{ATLANTA}, \texttt{GEORGIA}, \texttt{Atlanta}, c) > \Psi(\texttt{ATLANTA}, \texttt{OHIO}, \texttt{Atlanta}, c)$

- Entity type, as output by the named entity recognition system.

  $\Psi(\texttt{ATLANTA} - \texttt{CITY}, \texttt{Atlanta}, c) > \Psi(\texttt{ATLANTA} - \texttt{MAGAZINE}, \texttt{Atlanta}, c)$
  when tagged as LOCATION

co-reference resolution

Pronomial Reference

[PERSON **Boris Johnson**]'s cabinet is divided over how to proceed with Brexit, as the [PERSON **prime minister**] faces the stark choice of pressing ahead with **his** deal or gambling **his** premiership on a pre-Christmas general election. The [PERSON **prime minister**] told MPs at Wednesday's PMQs that **he** was awaiting the decision of the EU27 over whether to grant an extension before settling **his** next move. Some cabinet ministers, including the secretary, Julian Smith, believe the majority of 30 achieved by the government on the second reading of the Brexit bill on Tuesday suggests [PERSON **Johnson**]'s deal has enough support to carry it through all its stages in parliament.

Some Terminology

Referring expression Part of utterance used to identify or introduce an entity

Referents are such entities (imagined to be) in the world

Reference is the relation between a referring expression and a referent

Coreference More than one referring expression is used to refer to the same entity

Anaphora Reference to, or depending on, a previously introduced entity

Coreference and Pronouns

- Pronouns serve as anaphoric expressions when they rely on the previous discourse for their interpretation

  Definite pronouns `He`, `she`, `it`, `they`, etc.
  Indefinite pronouns `One`, `some`, `elsewhere`, `other`, etc.

Coreference and Pronouns

- Pronouns serve as anaphoric expressions when they rely on the previous discourse for their interpretation

  Definite pronouns `He, she, it, they,` etc.
  Indefinite pronouns `One, some, elsewhere, other,` etc.

- Some pronouns have other roles as well
  - periphrastic it: `It is raining`, `It is surprising that you ate a banana`
  - generic they and one: `They'll get you for that`, `One doesn't do that sort of thing in public`

Coreference and Pronouns

- Pronouns serve as anaphoric expressions when they rely on the previous discourse for their interpretation

  Definite pronouns `He, she, it, they,` etc.
  Indefinite pronouns `One, some, elsewhere, other,` etc.

- Some pronouns have other roles as well

  – periphrastic it: `It is raining`, `It is surprising that you ate a banana`
  – generic they and one: `They'll get you for that`, `One doesn't do that sort of thing in public`

- **Antecedent**: expression from the previous discourse used in interpreting a pronoun

Reference Resolution

- Reference resolution is the process of determining the referent of a referring expression

- Context obviously plays a crucial role in reference resolution

  Situational The real-world surroundings (physical and temporal) for the discourse
  Mental The knowledge/beliefs of the participants
  Discourse What has been communicated so far

Reference Resolution

- Reference resolution is the process of determining the referent of a referring expression

- Context obviously plays a crucial role in reference resolution

  Situational The real-world surroundings (physical and temporal) for the discourse
  Mental The knowledge/beliefs of the participants
  Discourse What has been communicated so far

- Most approaches to implementing reference resolution distinguish two stages

  1. Filter the set of possible referents by appeal to linguistic constraints
  2. Rank the resulting candidates based on some set of heuristics

Constraints on Pronouns: Feature Agreement

- English pronouns agree with number and/or gender of their antecedent

```
Robin has a new car.  It/*She/*They is red.
Robin has a sister.  *It/She/*They/*We is well-read.
Robin has three cars.  *It/*She/They/*We are all red.
```

Constraints on Pronouns: Feature Agreement

- English pronouns agree with number and/or gender of their antecedent

```
Robin has a new car.  It/*She/*They is red.
Robin has a sister.  *It/She/*They/*We is well-read.
Robin has three cars.  *It/*She/They/*We are all red.
```

- As well as the person (but case is determined locally):

```
Robin and I were late.  *Me/*They/We/I missed the show
Robin and I were late.  The usher wouldn't let *we/*I/us/me in.
```

Constraints on Pronouns: Feature Agreement

- English pronouns agree with number and/or gender of their antecedent

```
Robin has a new car.  It/*She/*They is red.
Robin has a sister.  *It/She/*They/*We is well-read.
Robin has three cars.  *It/*She/They/*We are all red.
```

- As well as the person (but case is determined locally):

```
Robin and I were late.  *Me/*They/We/I missed the show
Robin and I were late.  The usher wouldn't let *we/*I/us/me in.
```

- German pronouns agree with number and gender of antecedent

```
Hier ist ein Apfel.  Ich bedenke ob er/*sie/*es reif ist.  [masc.]
Here's an apple.  I wonder if *he/*she/it is ripe.  [neuter]
```

Constraints on Pronouns: Syntax

- When the text is in the same sentence, pronominal coreference is subject to binding conditions

  `Joe likes him` vs. `John likes himself`

  `Joe thinks Ann likes him/herself`
  vs. `*Joe thinks Ann likes himself`

  `Her brother admires Ann` Whose brother?

Constraints on Pronouns: Syntax

- When the text is in the same sentence, pronominal coreference is subject to binding conditions

  `Joe likes him` vs. `John likes himself`

  `Joe thinks Ann likes him/herself`
  vs. `*Joe thinks Ann likes himself`

  `Her brother admires Ann` Whose brother?

- And, sometimes, to selectional restrictions based on the verb that governs it

  `Joe parks his car in the garage. He has driven it around for hours`

  it = the car, it ≠ garage

  `I picked up the book and sat in a chair. It broke`

  it = chair, it ≠ book

Constraints Not Enough

- The kind of strong constraints we've just seen are not always enough to reduce the candidate set for resolution to a single entity

```
John punched Bill.  He broke his jaw
John punched Bill.  He broke his hand

Tom hates her husband, but Jane worked for him anyway
Tom hates her husband, but Jane stays with him anyway
```

Heuristics for Pronoun Interpretation

- Many different features influence how a listener will resolve a definite pronoun (i.e., what they will take to be its antecedent)

Recency The most recently introduced entity is a better candidate

```
First Robin bought a phone, and then a tablet.  Kim is always
borrowing it
```

Heuristics for Pronoun Interpretation

- Many different features influence how a listener will resolve a definite pronoun (i.e., what they will take to be its antecedent)

Recency The most recently introduced entity is a better candidate

`First Robin bought a phone, and then a tablet. Kim is always borrowing it`

Grammatical role Some grammatical roles (e.g. SUBJECT) are felt to be more salient than others (e.g., OBJECT)

`Bill went to the pub with John. He bought the first round`

`John` is more recent, but `Bill` is more salient.

Heuristics

Repeated mention A repeatedly-mentioned entity is likely to be mentioned again

```
John needed portable web access for his new job.  He decided he
wanted something classy.  Bill went to the Apple store with him.
He bought an iPad.
```

`Bill` is the previous subject, but `John`'s repeated mentions tips the balance.

Heuristics

Repeated mention A repeatedly-mentioned entity is likely to be mentioned again

```
John needed portable web access for his new job.  He decided he
wanted something classy.  Bill went to the Apple store with him.
He bought an iPad.
```

`Bill` is the previous subject, but `John`'s repeated mentions tips the balance.

Parallelism Parallel syntactic constructs can create an expectation of coreference in parallel positions

```
Susan went with Ann to the cinema.  Carol went with her to the pub
```

Heuristics

Verb semantics A verb may serve to foreground one of its argument positions for subsequent reference because of its semantics

`John criticized Bill after he broke his promise vs. John telephoned Bill after he broke his promise`

`Louise apologized to/praised Sandra because she ...`

Heuristics

Verb semantics A verb may serve to foreground one of its argument positions for subsequent reference because of its semantics

`John criticized Bill after he broke his promise vs. John telephoned Bill after he broke his promise`

`Louise apologized to/praised Sandra because she ...`

World knowledge At the end of the day, sometimes only one reading makes sense

`The city council denied the demonstrators a permit because they feared violence`

vs.

`The city council denied the demonstrators a permit because they advocated violence`

Automatic Methods

- Rich history of automatic definite reference and pronoun resolution systems
  - initially rule-based
  - more recently using machine learning
- Viewed as a simple binomial classification task
  - for every pair of referring expressions
  - are they coreferential, or not?

Supervised Training

- Given an corpus annotated for coreference, to train a model we simply
  - given an NP$_k$ that is known to co-refer with NP$_j$ where NP$_j$ is the closest such NP, create a positive training instance (NP$_k$, NP$_j$).
  - for all NPs between NP$_k$ and NP$_j$, create a negative training instance(NP$_k$, NP$_{j+1}$), (NP$_k$, NP$_{j+2}$), etc.

Supervised Training

- Given an corpus annotated for coreference, to train a model we simply
  - given an NP_k that is known to co-refer with NP_j where NP_j is the closest such NP, create a positive training instance (NP_k, NP_j).
  - for all NPs between NP_k and NP_j, create a negative training instance(NP_k, NP_{j+1}), (NP_k, NP_{j+2}), etc.

- Tabulate the value of likely candidate features,

Supervised Training

- Given an corpus annotated for coreference, to train a model we simply
  - given an NP_k that is known to co-refer with NP_j where NP_j is the closest such NP, create a positive training instance (NP_k, NP_j).
  - for all NPs between NP_k and NP_j, create a negative training instance(NP_k, NP_{j+1}), (NP_k, NP_{j+2}), etc.

- Tabulate the value of likely candidate features,such as
  - the nature of NP_k and NP_j: pronouns, definite NPs, demonstrative NPs (this/that/these/ those X), proper names;

Supervised Training

- Given an corpus annotated for coreference, to train a model we simply
  - given an NP_k that is known to co-refer with NP_j where NP_j is the closest such NP, create a positive training instance (NP_k, NP_j).
  - for all NPs between NP_k and NP_j, create a negative training instance(NP_k, NP_{j+1}), (NP_k, NP_{j+2}), etc.

- Tabulate the value of likely candidate features,such as
  - the nature of NP_k and NP_j: pronouns, definite NPs, demonstrative NPs (this/that/these/ those X), proper names;
  - distance between NP_k and NP_j: 0 if same sentence, 1 if adjacent sentence, etc.;

Supervised Training

- Given an corpus annotated for coreference, to train a model we simply
  - given an NP_k that is known to co-refer with NP_j where NP_j is the closest such NP, create a positive training instance (NP_k, NP_j).
  - for all NPs between NP_k and NP_j, create a negative training instance(NP_k, NP_{j+1}), (NP_k, NP_{j+2}), etc.

- Tabulate the value of likely candidate features,such as
  - the nature of NP_k and NP_j: pronouns, definite NPs, demonstrative NPs (this/that/these/ those X), proper names;
  - distance between NP_k and NP_j: 0 if same sentence, 1 if adjacent sentence, etc.;
  - whether NP_k and NP_j agree in number;

Supervised Training

- Given an corpus annotated for coreference, to train a model we simply
  - given an NP$_k$ that is known to co-refer with NP$_j$ where NP$_j$ is the closest such NP, create a positive training instance (NP$_k$, NP$_j$).
  - for all NPs between NP$_k$ and NP$_j$, create a negative training instance(NP$_k$, NP$_{j+1}$), (NP$_k$, NP$_{j+2}$), etc.

- Tabulate the value of likely candidate features,such as
  - the nature of NP$_k$ and NP$_j$: pronouns, definite NPs, demonstrative NPs (this/that/these/ those X), proper names;
  - distance betwen NP$_k$ and NP$_j$: 0 if same sentence, 1 if adjacent sentence, etc.;
  - whether NP$_k$ and NP$_j$ agree in number;
  - whether NP$_k$ and NP$_j$ agree in gender;

Supervised Training

- Given an corpus annotated for coreference, to train a model we simply
  - given an NP_k that is known to co-refer with NP_j where NP_j is the closest such NP, create a positive training instance (NP_k, NP_j).
  - for all NPs between NP_k and NP_j, create a negative training instance(NP_k, NP_{j+1}), (NP_k, NP_{j+2}), etc.

- Tabulate the value of likely candidate features,such as
  - the nature of NP_k and NP_j: pronouns, definite NPs, demonstrative NPs (this/that/these/ those X), proper names;
  - distance between NP_k and NP_j: 0 if same sentence, 1 if adjacent sentence, etc.;
  - whether NP_k and NP_j agree in number;
  - whether NP_k and NP_j agree in gender;
  - whether their semantic classes are in agreement;

Supervised Training

- Given an corpus annotated for coreference, to train a model we simply
  - given an NP_k that is known to co-refer with NP_j where NP_j is the closest such NP, create a positive training instance (NP_k, NP_j).
  - for all NPs between NP_k and NP_j, create a negative training instance(NP_k, NP_{j+1}), (NP_k, NP_{j+2}), etc.

- Tabulate the value of likely candidate features,such as
  - the nature of NP_k and NP_j: pronouns, definite NPs, demonstrative NPs (this/that/these/ those X), proper names;
  - distance betwen NP_k and NP_j: 0 if same sentence, 1 if adjacent sentence, etc.;
  - whether NP_k and NP_j agree in number;
  - whether NP_k and NP_j agree in gender;
  - whether their semantic classes are in agreement;
  - edit distance between NP_k and NP_j;

- Use any supervised learning method to train a model

relation extraction

Relations

- We may be interested in relations of a specific type

- Example: birthplaces

```
Bill Clinton was born in the small town of Hope, Arkansas, ...
George Walker Bush was born in New Haven, Connecticut, while ...
Obama was born in Hawaii, studied at Columbia and Harvard, ...
```

- Broad category: ENTITY-ORIGIN

Types of Relations

- Types of relations from SemEval-2010

CAUSE-EFFECT	those cancers were caused by radiation exposures
INSTRUMENT-AGENCY	phone operator
PRODUCT-PRODUCER	a factory manufactures suits
CONTENT-CONTAINER	a bottle of honey was weighed
ENTITY-ORIGIN	letters from foreign countries
ENTITY-DESTINATION	the boy went to bed
COMPONENT-WHOLE	my apartment has a large kitchen
MEMBER-COLLECTION	there are many trees in the forest
COMMUNICATION-TOPIC	the lecture was about semantics

Pattern-Based Relation Extraction

- Surface patterns

```
[PERSON] was born in [LOCATION]
```

- Not robust to small variations

```
Bill Clinton was born in the small town of Hope, Arkansas, ...
Ronald Reagan who was born in Tampico, Illinois ...
Jimmy Carter was born October 1, 1924 in Plains, GA.
```

- Possibly many patterns needed
  - hand-crafted patterns likely high precision, low recall
  - learned patterns require annotated training data or known examples

Syntactic Patterns

- Patterns can be also defined over the syntactic relations

- Dependency relationships

  `[PERSON]` $\leftarrow$`SUBJ`— `born` —`PP-LOC`$\rightarrow$ `[LOCATION]`

- Recall: semantic roles

Learning Patterns

- Given a set of examples for a relation `ENTITY-ORIGIN(person, location)`
- Automatically label text where both `person` and `location` occur
- Use this as training data to learn classifier
- Features
  - properties of the entities
  - words and n-gram between and around entities
  - syntactic dependency path between entities

knowledge base population

Wikipedia Infobox

- Given a frame
- Slot filling
- Each slot a relation
- Possibly multiple entries in a slot (children, education)

Jimmy Carter

39th President of the United States

In office
January 20, 1977 – January 20, 1981

Vice President	Walter Mondale
Preceded by	Gerald Ford
Succeeded by	Ronald Reagan

Personal details

Born	James Earl Carter Jr. October 1, 1924 (age 95) Plains, Georgia, U.S.
Political party	Democratic
Spouse(s)	Rosalynn Smith (m. 1946)
Children	Jack · James III (Chip) · Donnel (Jeff) · Amy
Relatives	James Earl Carter Sr. (father) Lillian Gordy (mother)
Residence	Plains, Georgia, U.S.
Education	Georgia Institute of Technology United States Naval Academy (BS)

Information Fusion

Born	James Earl Carter Jr. October 1, 1924 (age 95) Plains, Georgia, U.S.

- Combine information from multiple text sources

```
Jimmy Carter celebrates his birthday today on October 1.
Born in 1924, Carter is the oldest president alive, ...
President Carter who hails from Plains, Georga, ...
```

Events

- Events involve multiple relations

- Limited to a specific time frame

- Event co-reference
  - multiple mentions of same event
  - mention same time frame, same actors, etc.
  - clustering? linking?

- Relations between events
  - temporal relationships
  - causal relationships

Hedges, Denials, Hypothetical

- Examples

1. `GM will lay off workers.`
2. `A spokesman for GM said GM will lay off workers.`
3. `GM may lay off workers.`
4. `The politician claimed that GM will lay off workers.`
5. `Some wish GM would lay off workers.`
6. `Will GM lay off workers?`
7. `Many wonder whether GM will lay off workers.`
8. `This suggests that GM will lay off workers.`

- Probability of proposition (`may`)
- Hedging (`suggests`)
- Attribution (`spokesman said`, `politician claimed`)

questions?